



# SPELLCHECKER

Подготвила  
студентка гр. БКЛ-194  
Симонова Татяна

5 марта

Добрый день! Вот мои наброски по проекту.

план минимум1: создать программу, которая будет на вход принимать слово и проверять его на орфографию. Если слово написано правильно, возвращать `correctly`, а если непраивльно - верный вариант

план максимум: создать опечаточник, который будет исправлять опечатки в большом тексте. Будет принимать на вход текст, а на выход возвращать новый файл с исправленными словами.



## STEPS

АНАЛИЗ  
ЛИТЕРАТУРЫ

НАПИСАНИЕ КОДА  
ДЛЯ РАБОТЫ  
SPELLCHECKER

НАПИСАНИЕ КОДА  
ДЛЯ ДИЗАЙНА  
ПРОГРАММЫ



# ПОСЛЕ 2,5 МЕСЯЦЕВ РАБОТЫ

ПЛАН МИНИМУМ ✓

ПЛАН МАКСИМУМ (МОДИФИЦИРОВАН)

В ходе работы выяснилось, что в силу несовершенства частотных словарей опечатки исправляются не всегда корректно.

Записывать новый файл с исправленным текстом оказалось бессмысленным. Поэтому механическое исправление ошибок предоставлено самому пользователю. Для каждого неверно написанного слова программа выводит на экран запись "Возможно, вы имели в виду не **wrongword**, а **correctword**".

С помощью `ctrl+C`, `ctrl+F`, `ctrl+V` пользователь сможет самостоятельно исправить ошибки, если это потребуется.

ПЛАН МАКСИМУМ ++

Мне захотелось, чтобы программа была удобна в использовании, поэтому решила написать для нее графический интерфейс с помощью модуля Tkinter.

```
import Counter
```



Добро пожаловать в приложение PythonGrammer



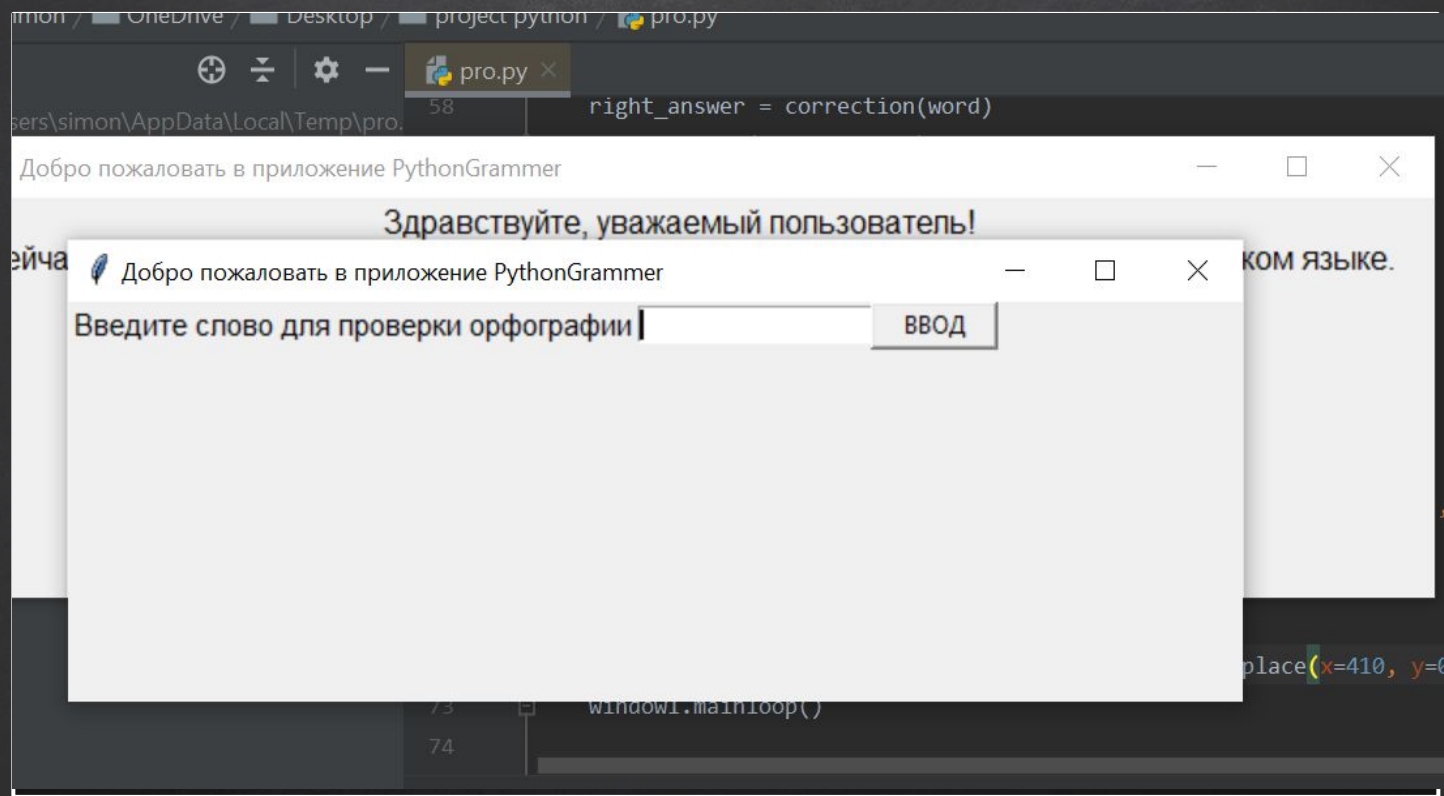
Здравствуйте, уважаемый пользователь!

Сейчас вы используете программу по поиску орфографических ошибок в словах на русском языке.  
Где бы вы хотели искать ошибки? Выберите нужный вариант.

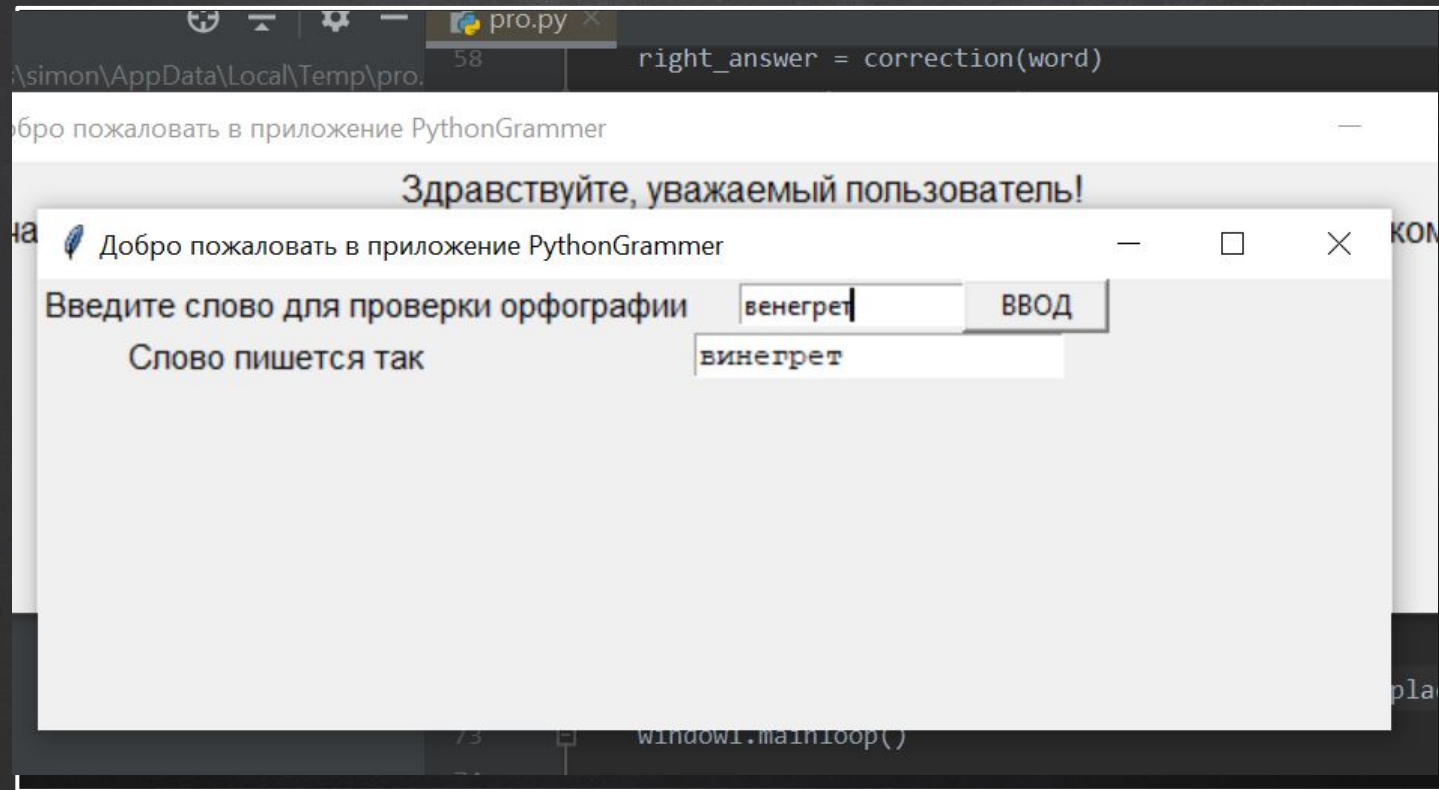
В словах

В тексте

```
Python %s on %s' % (sys.version, sys.platform)
```







avigate Code Refactor Run Tools VCS Window Help pro.py [C:\Users\simon\AppData\Local\Temp\pro.py1] - C:\Users\simon\O

on > OneDrive > Desktop > project python > pro.py

pro.py x  
58 right\_answer = correction(word)

добро пожаловать в приложение PythonGrammar

Добро пожаловать в приложение PythonGrammar

Введите название вашего файла с расширением proba.txt

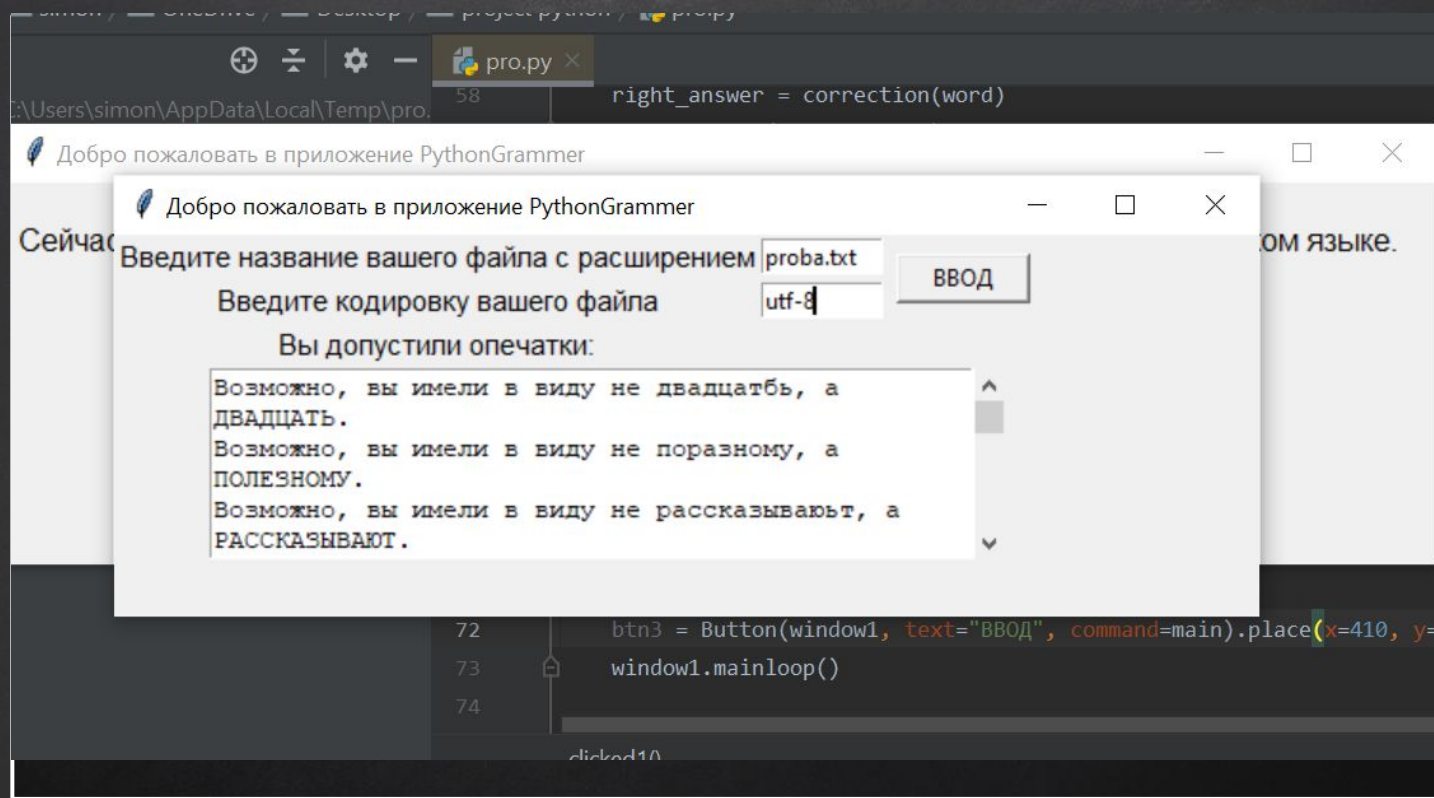
Введите кодировку вашего файла

utf-8

ВВОД

```
72 btn3 = Button(window1, text="ВВОД", command=main).place(x=410, y=0, w  
73 window1.mainloop()
```





# Как это работает?

Программа старается подобрать слово, которое является наиболее вероятным вариантом написания того, что мы ввели с клавиатуры (или которое есть в нашем тексте).

“Комбинированная вероятность” складывается из:

- вероятности употребления слова в русском тексте (для этого нам нужен частотный словарь; я брала с [pymorphy](#))
- вероятности того, может ли то или иное слово быть напечатано, когда автор имел другое (для этого прописываются и запоминаются все возможные варианты ошибок для слова: пропуск буквы, замена буквы, вставка дополнительной буквы и т.д.)

Из всех кандидатов выводится то слово, “комбинированная вероятность” которого наибольшая

За основу этого проекта была взята статья <https://norvig.com/spell-correct.html>



## TROUBLES

Основные сложности возникли при соединении кода для интерфейса и кода spellchecker - return переменной в функциях не срабатывал из-за появившихся кнопок с командами.

Какой выход?

Метод `StringVar()`

Он нужен для того, чтобы отслеживать изменения в переменных tkinter.

```

def main():
    word = ss.get()
    right_answer = correction(word)
    print_answer(right_answer)

def clicked1():
    global window1
    window1 = Toplevel(window)
    window1.overridereDIRECT = True
    window1.geometry('600x200')
    z = Entry(window1, width=20, textvariable=ss)
    lbl = Label(window1, text='Введите слово для проверки орфографии', font=("Arial Bold", 11))
    lbl.grid(column=0, row=0)
    z.grid(column=1, row=0)
    z.focus()
    btn3 = Button(window1, text="ВВОД", command=main).place(x=410, y=0, width=65, height=24)
    window1.mainloop()

```

```

ss = StringVar()
ss.set("")

```



THANKS!

Any questions?

You can find my project on

<https://github.com/tanyuhich/project>